BurdettML Vs WS Chor Requirements Comparison

Area	Ref	Description	Comment
Charter Constraints			
	D-CR-001	All specified choreography descriptions MUST be compatible with WSDL 1.2.	NA. The spec defines an abstract choreography. A separate binding is required to map to WSDL
	D-CR-014	A choreography SHOULD not be bound to any specific implementation.	Met, The spec defines an abstract choreography and so is implementation independent
Interoperability			
	D-CR-010	A choreography MAY provide an extensible binding mechanism such that choreographies could be bound to diverse technologies.	NA. The spec defines an abstract choreography. However it is designed to be used with a binding mechanism but does not define one
	D-CR-023	A choreography definition language MUST NOT be restricted to any single implementation.	Met. The spec defines an abstract choreography and so is implementation independent
Management & Provisioning			
	D-CR-004	It MUST be possible to query the state of a choreography.	Not met. The spec specifies the states that may exist when performing a choreography but does not specify how to query them.
	D-CR-052	It MUST be possible to manage choreographies and their relationships and the messages exchanged between them.	Don't know. What is meant by "manage" in this context?
	D-CR-053	It MUST be possible to observe the state of a choreography as it occurs in time.	Not met. The spec specifies the states that may exist when performing a choreography but does not specify how to observe them.
	D-CR-056	It MUST be possible to create "limited global views" of a choreography, where not all interactions are presented.	Not met. The spec allows choreographies to be extended but does not define how to create restricted views.
	D-CR-013	A choreography MUST support for a specific observer role.	Not met. There is no separate observer role
	D-CR-016	A choreography MUST provide a global model for presenting its interactions from the point of view of all the parties and not from the point of view of just one party.	Met. Fully supported.
	D-CR-020	It MUST be possible and practicable to store instances of use of choreographies in a repository, and to be a able to search for such instances and to retrieve them.	Not met. The spec defines an abstract choreography and does describe or define *instances* of choreographies or how to manage or use them.
Exception Handling			
	D-CR-005	A choreography MUST provide exception handling that allows propagation of errors.	Partially met. The spec allows definitions of states that are error states but does not define how those states are propagated
	D-CR-009	A choreography MUST provide the ability to transition to a distinct state when a timeout occurs.	Partially met. The spec would allow the specification choreographies with states that corresponded to a timeout.
	D-CR-018	A choreography SHOULD define a standardized way to recover from catastrophic failure.	Not met. Although a choreography could be defined that allowed attempts at a recovery from a catastrophic failure.
	D-CR-026	A choreography MUST support robust exception handling.	Not met - although there has been some discussion on the list on how it could be extended to do this.
	D-CR-027	Error/fault handling and compensation features MUST to be able to be expressed in the choreography definition language.	Met. The spec allows states to be defined that correspond to errors or faults. The existence of these states allows the definition of the compensation interactions and messages required.
	D-CR-030	A choreography MUST be able to distinguish error and regular state transitions, and to describe unhandled exceptions.	Partially met. Only met in the sense that semantic definitions associated with a state can identify them as regular or error. There is no attribute that identifies this.
	D-CR-031	A choreography MUST support an exception condition for Delivery Failure - The message was sent but was (probably) not received.	Partially met. Only met in the sense that semantic definitions associated with a state can identify them as regular or error. There is no attribute that identifies this.
	D-CR-032	It MUST be possible to manage exceptions between choreographies, and to include exception handling information in dependency management.	Don't know. What is meant by "exceptions between choreographies" in this context?
	D-CR-033	A choreography MUST provide support for defining behavior of the system, when valid error/exception messages arrive for choreography instances after their completion (or before their initiation).	Met. The spec would allow the definition of a condition where a message was received after the instance was complete. Interactions/messages could then be defined that would follow.
	D-CR-043	It MUST be possible to differentiate errors (unknown and fatal) and exceptions (known and potentially recoverable) in the context of choreography.	Met. Exceptions and errors are defined as a state and a state is completely flexible in its definition and meaning.

BurdettML Vs WS Chor Requirements Comparison

Area	Ref	Description	Comment
	D-CR-061	A choreography MUST provide the ability to transition to a predefined state when an exception occurs.	Met. The spec would allow the definition of a choreography that transitioned to a state when an exception occurred.
	D-CR-064	If a process detects that a choreography is not being followed correctly, then the process SHOULD be able to use the choreography definition to identify exactly what went wrong.	Met. The spec is designed so that the choreography definition can be used for this purpose
	D-CR-065	A choreography definition language MUST support the following exception: message format error - the components of the message were not validly constructed.	Met. The spec allows the specification of a state that corresponds to this.
	D-CR-066	A choreography definition language MUST support the following exception: message Content structure error.	Met. See response to D-CR-065
	D-CR-067	A choreography definition language SHOULD support exceptions whereby a given service component could be sent at any point in the transaction.	Met. The spec is completely flexible in terms of the conditions that apply when sending any message.
	D-CR-068	Exception types MUST be limited to the ones that will be used in the choreography definition language.	Partially met. Different exception types can be defined as states. However, standard exception types are not defined.
	D-CR-069	It MUST be possible to model different states for termination of the choreography (e.g. failure and success).	Met. Although the semantic definitions are used to identify which represent failure and which success.
	D-CR-070	A choreography definition language MUST support the following exception: transmission error - message not sent.	Met. See response to D-CR-065
Messaging and Protocols			
	D-CR-002	A choreography MUST be independent of message formats.	Met. Fully supported.
	D-CR-007	A choreography SHOULD express the types of messages a participant may send, and the types of messages/responses the participant should anticipate receiving from the other participant(s) (including time-outs) based on the apparent state of the exchange.	Met. Fully supported.
	D-CR-011	A choreography SHOULD describe exchanges of information that change the state of the process.	Met. Fully supported.
	D-CR-029	A choreography MUST support some standard taxonomy of messages, such as a business messages, generic error reporting messages and acknowledgement messages.	Not met. There is no taxonomy of message types as all messages are treated equally although all the different types of messages could be defined.
	D-CR-035	It MUST be possible to model message flows that repeat, based on information within the messages (for instance, the contract negotiation protocol).	Met. Looping of messages is possible
	D-CR-039	It MUST be possible to describe a sequence of communications among choreographies.	Met. Sequences of messages can be defined
	D-CR-051	A choreography definition language MUST provide a construct that describes the sending of a single message.	Met. A choreography consisting of a single message could be defined
	D-CR-055	It SHOULD be possible to define a choreography the uses a callback mechanism.	Not met. There is no direct support for call back although it could be included in a binding of a choreography to an implementation
	D-CR-060	It MUST be possible to define choreography without having to specify the contents of the messages being used.	Met. Fully supported using Message Families
Interfaces			
	D-CR-015	A choreography MUST provide the ability to have prose associated with it to enable its behaviour to be explained.	Met. Many different parts of the choreography can have text in multiple languages defined to describe the semantics.
	D-CR-044	A choreography MUST enable information hiding.	Don't know. What is "information hiding" in this context?
Transaction			
	D-CR-017	A choreography SHOULD enable a agreement or agreement(s) that provides the business context of the choreography definition.	Partially met. Business context could be included in the definition of a state that triggers the processing of the different parts of the choreography.
	D-CR-037	It MUST be possible to describe a behaviour recursively.	Not met, as there is no composition facility defined.
	D-CR-038	It MUST be possible to describe conditional behaviour for a choreography.	Met, using the "precondition" on each interaction
	D-CR-048	It MUST be possible to describe a choreography in terms of its messaging behaviour.	Met, the spec defines messaging behaviour

BurdettML Vs WS Chor Requirements Comparison

Area	Ref	Description	Comment
	D-CR-057	It MUST be possible to define multi-party interaction.	Met, there can be any number of roles specified, each of which can be mapped to a different party
	D-CR-058	It MUST be possible for a choreography to modify its behaviour based on its operational context.	Met, this can be met by defining the semantics associated with states.
	D-CR-059	It MUST be possible to describe negotiation between web services participating in a choreography, where the result of a transaction may depend on repeated iterations of an ask/answer cycle.	Met. A choreography with these characteristics could be defined
Composition			
	D-CR-003	A choreography MAY have run time changes which allow the actual participants to vary.	Partially met. The spec only defines an abstract choreography. A binding could probably created that allowed this type of behavior.
	D-CR-006	It MUST be possible to describe choreographies as a composition of other choreographies.	Not met. There is no composition capability.
	D-CR-019	It MUST be possible to make a choreography C2 dependent on another choreography C1 such that you can only create a new instance of C2 after a related instance of C1 has been completed.	Met. You can specify that the following of one choreography is dependent on the following of an earlier one.
	D-CR-024	It MUST be possible to dynamically determine the participants in a choreography at runtime.	Partially met. The spec only defines an abstract choreography. A binding could probably created that allowed this type of behavior.
	D-CR-034	Choreographies MUST be composable into a hierarchy.	Not met. There is no composition capability.
	D-CR-036	Choreographies SHOULD be able to call other choreographies in a hierarchical fashion.	Not met. There is no composition capability.
	D-CR-040	It MUST be possible to describe parallel composition of services.	Not met. There is no composition capability.
	D-CR-041	It MUST be possible to model events that are strictly related in time, as well as those that are unrelated in time - i.e. parallelism or partial ordering.	Met. Messages can occur in parallel or partially ordered.
	D-CR-042	A choreography MAY have run time changes which allow the behaviour of the actual choreography to vary based on state.	Met. The spec defines states that can be used to control the choreography being followed
	D-CR-047	It MUST be possible to define a new choreography by "extending" an existing one.	Met. The spec defines how to extend a choreography
	D-CR-054	A choreography SHOULD express the composition of participant web services into a new service.	Not met. The spec defines an abstract choreography - a binding to a web service would be required to realize this.
	D-CR-062	It MUST be possible to dynamically add sub-choreographies to a "running" choreography.	Not met. However I think this is a "business process" rather than a choreography problem
Testing and Validation			
	D-CR-008	An implementation of a process that is following a choreography MUST be able to verify that the choreography is being followed correctly as specified in the choreography definition.	Met. The definitions produced according to the spec are designed to be used so that checking that a choreography is being followed correctly is possible.
	D-CR-049	It MUST be possible to validate a choreography definition for correct behaviour at the time it is designed.	Don't know. This can only be done by a review process. There problem is that in order to do this you already need a definition of "correct behavior" however "correct behavior is what the choreography definition is supposed to define.
	D-CR-050	It MUST be possible to validate a choreography definition for correct behaviour at the time it is in operation.	Met. The spec is defined to support this.
Ease of Use			Met. Each choreography definition is given a URI.
	D-CR-012	A choreography MUST be uniquely named.	
Support for Semantics			
	D-CR-025	There SHOULD be a distinction between a "participant" and a "role", where the participants might be dynamic but the roles need not be.	Met. The spec just defines roles. Participants could be defined in the binding